MASSA
GENERATIONS AHEAD IN SONAR & ULTRASONIC TECHNOLOGY

MASSA PRODUCTS CORPORATION
280 LINCOLN STREET, HINGHAM, MA 02043-1796 U.S.A.
781-749-4800 FAX: 781-740-2045 www.massa.com

Massa Model M3 Wireless Tank Level Sensor and Digi Gateway User's Guide

.

VYDAS
Vydas International Marketing
Swan House PBC Lynchborough Rd
Passfield Hampshire GU30 7SB UK

Tel: 44(0)1428 751822 Fax: 44(0)1428 751833
Email: info@vydas.co.uk Web: www.vydas.co.uk

Table of Contents

Section		Page
1.0	Overview and General Operation	1
2.0	Gateway Basic Settings	2
3.0	Operating Modes for Gateway and Sensor	5
	- Configuring Sensor	5
	- Normal Operation	11
	- Diagnostic Mode – Waveforms	12
	- Firmware Update	13
	- M3 Application Software enable (X4 Gateway only)	13
	- Sensor XBee Firmware Update	14
4.0	Application Setup Example	15
Appendix A	Miscellaneous Information	18
	- Battery Life and Suggested Replacement Voltage	
	- Diagnostic Mode – Sensor Waveforms	
	- LED Behavior	
	- Rebooting Sensor Manually	
	- Firmware Releases	
Appendix B	dia.yml format	21
Appendix C	Digi Gateway & Massa M3 Sensor Placement Strategies	22
Appendix D	Fresnel Zone	24
Appendix E	Outgoing Auto-Message Message Format	25
Appendix F	M3 Application Software – Understanding an Ultrasonic Waveform	26
Appendix G	Firmware Conversion Procedure and Checklist for Reliable Sensor and Gateway Operation	27
Appendix H	Troubleshooting Guide	28

North America

The M3 Sensor complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (*i.*) this device may not cause harmful interference and (*ii.*) this device must accept any interference received, including interference that may cause undesired operation. Contains FCC ID: MCQ-XBEEPRO2, IC: 1846A-XBEEPRO2

Europe

The M3/150CE and M3/95CE Sensors comply with European Council EMC Directive 2004/108/EC (EMC) and Low Voltage Directive 2006/95/EC (LVD).

Manufactured under ISO 9001:2008 registered standards

If the equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

Massa Products Corporation provides this document "as is," without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose. Massa Products Corporation may make improvements and/or changes in this manual at any time. This product could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes may be incorporated in new editions of the publication.

1.0 Overview and General Operation

The Massa Model M-3 Wireless Ultrasonic Sensor communicates its tank level information through a Digi International ConnectPort Gateway. This gateway requires set up that is described in this manual. Sensors from the factory are programmed to stay awake and not set up for automated messages. There are several operating modes using the Massa Python code to control the gateway basic functions. This includes setting sensor settings for the application (range/status acquisition rate, sleep period, etc), normal monitoring mode (listen only to sensor's automated messages, diagnostic mode, and firmware programming.

The gateway typically is set up operate with up to 12 available RF channels (SC=0x1ffe). When the sensor power's up, it will first search for the gateway's operating channel and attempt will join it if the PAN ID matches and if security key matches (if enabled). Once a sensor is associated with a gateway, it will remain associated even during its sleep period, as long as the gateway's poll timeout does not expire (poll timeout is reset when sensor wakes up). If the poll timeout does expire or gateway power cycle causes a selection of a new RF channel, then the sensor will rejoin upon the next wakeup cycle (ver 29 or greater).

2.0 Gateway Basic Settings

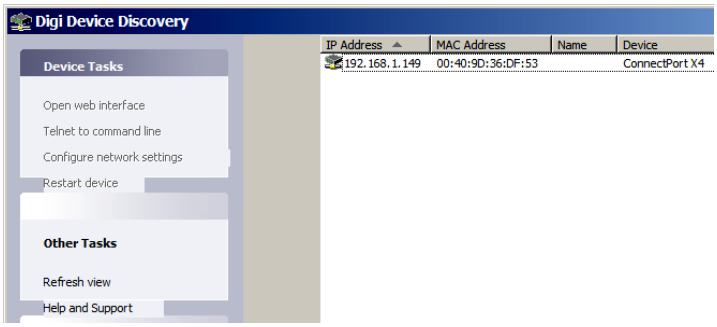
The Digi International Gateway, models X2 or X4, needs to be set up to be used with the M3 Sensor. This includes configuring for an Ethernet or WiFi connection, configuring gateway's XBee settings, loading python programs, and configuring server connection to receive the data. Most of the information can be found on Digi's web site (digi.com) on how to set up the gateway's Ethernet connection and server connection. The settings listed in this document are recommended for most applications.

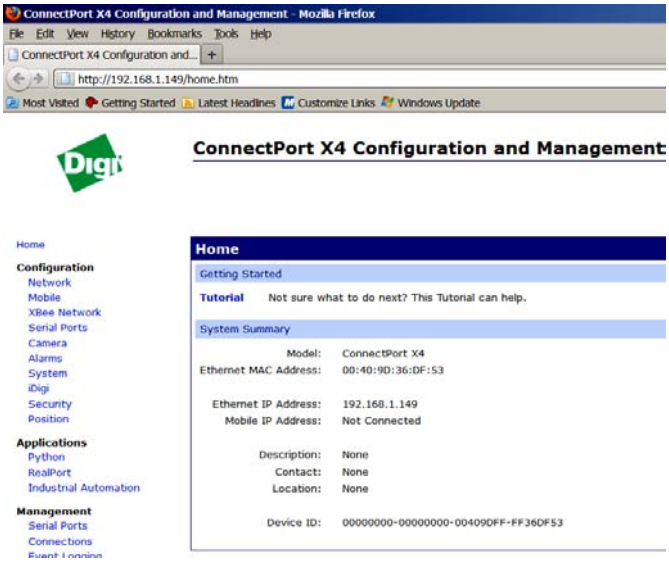
Basic Configuration Checklist

- ☐ Network Settings
- ☐ Time source
- ☐ XBee SP SN settings
- ☐ XBee PAN ID
- ☐ Server Connection (not necessary for initial evaluation)

2.1 Network Settings

Gaining access to the gateway requires an Ethernet connection and the use of Digi's Device Discovery application software. The gateway's Ethernet IP Settings are configured from the factory to *obtain an IP address automatically using DHCP*. Contact your network administrator if you need support for this. Initially, you may plug the gateway directly into your PC Ethernet port with your PC's Local Area Connection TCP/IP set to obtain an IP address automatically. Run the Device Discovery to find the gateway (see below). Double click the IP address found to get to the Digi Web UI (see lower figure).

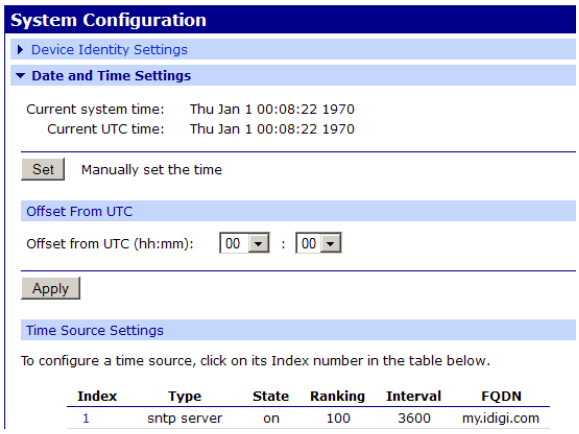


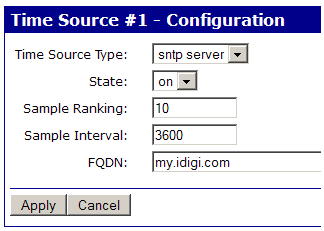


2.0 Gateway Basic Settings

2.2 Time Source

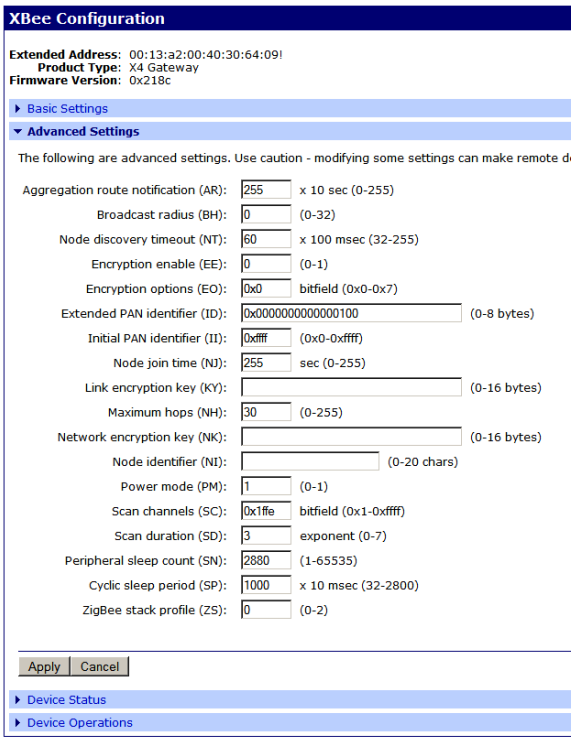
Initially you may set up the Time Source by manually entering the time which will be retained up to 10 days when powered off (model X4 gateways only). Go to the *System Configuration – Date and Time Settings* Web UI and initially, set the time manually. Next, in the *Time Source Settings* UI, you may select a time source such as my.idigi.com or other reliable time sources.


System Configuration
Device Identity Settings
Date and Time Settings
Current system time: Thu Jan 1 00:08:22 1970
Current UTC time: Thu Jan 1 00:08:22 1970
Set
Manually set the time
Offset From UTC
Offset from UTC (hh:mm):
00
00
Apply
Time Source Settings
To configure a time source, click on its Index number in the table below.
Index Type State Ranking Interval FQDN
1 sntp server on 100 3600 my.idigi.com



2.3 Gateway's XBee Settings

The settings listed below are ones that should be adjusted from the factory default settings. The Web UI for these settings can be found at *XBee Network*, and then clicking on the gateway's network or extended address. Select *Advanced Settings* to program these registers below.


XBee Configuration
Extended Address: 00:13:a2:00:40:30:64:09!
Product Type: X4 Gateway
Firmware Version: 0x218c
Basic Settings
Advanced Settings
The following are advanced settings. Use caution - modifying some settings can make remote d
Aggregation route notification (AR): 255 x 10 sec (0-255)
Broadcast radius (BH): 0 (0-32)
Node discovery timeout (NT): 60 x 100 msec (32-255)
Encryption enable (EE): 0 (0-1)
Encryption options (EO): 0x0 bitfield (0x0-0x7)
Extended PAN identifier (ID): 0x0000000000000100 (0-8 bytes)
Initial PAN identifier (II): 0xffff (0x0-0xffff)
Node join time (NJ): 255 sec (0-255)
Link encryption key (KY): (0-16 bytes)
Maximum hops (NH): 30 (0-255)
Network encryption key (NK): (0-16 bytes)
Node identifier (NI): (0-20 chars)
Power mode (PM): 1 (0-1)
Scan channels (SC): 0x1ffe bitfield (0x1-0xffff)
Scan duration (SD): 3 exponent (0-7)
Peripheral sleep count (SN): 2880 (1-65535)
Cyclic sleep period (SP): 1000 x 10 msec (32-2800)
ZigBee stack profile (ZS): 0 (0-2)
Apply
Cancel
Device Status
Device Operations

- ☐ Extended PAN ID (ID): xxx
  Gateway XBee PAN ID = 0 will assign a random value and sensors must have PAN ID=0 to associate. If multiple gateways are placed in the vicinity of each other, then the user should assign unique non-zero identical PAN IDs for each application. See section 3.1.14 for more information.
- ☐ Scan Channels Scan Channels (SC): 0x1ffe
  The SC register from a factory shipped gateway is 0x3fff or 0x7fff. Sensors are programmed are defaulted to 0x1ffe. It is recommended that the gateway be set to 0x1ffe.
- ☐ Peripheral sleep count (SN): see below
- ☐ Cyclic sleep period (SP): 1000 (recommended)
  The SN and SP registers control the poll timeout for the gateway which holds the sensor in its child table. The formula is:

  Poll Timeout = 0.03 x SN x SP (sec)

  For example, to retain the sensor for 24 hours (86,400 seconds) with a SP recommended setting of 1000: SN = 86,400 ÷ 30 seconds =2880. Every time the sensor wakes up from sleep and commissions to the gateway, it will reset its own poll timeout timer. A gateway will drop a sensor if it does not get a response from the sensor for the entire poll timeout period (for example a truck has blocked the RF signal). User may want to program this Poll Timeout setting to at least 4 times the sensor's sleep timer setting (see Section 3.1.5). Other applications where there can be more than 10 sensors on a gateway (limit 10), the Poll Timeout can be set to intentionally drop sensors so others can join. Recommended setting for such an application is 1 minute. Note that sensors will be required to rejoin each time awake cycle, which will slightly use more battery power in this type of application.

Click the Apply button to accept all the settings.

2.0 Gateway Basic Settings

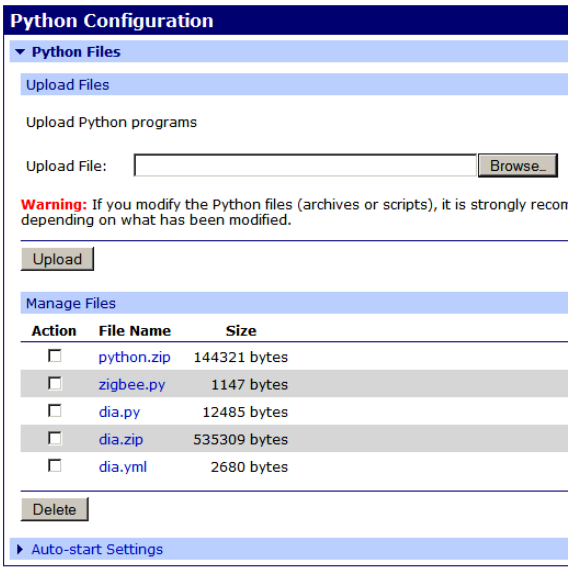
2.3 Gateway's XBee Settings (continued from previous page)

2.4 Gateway's Python Files

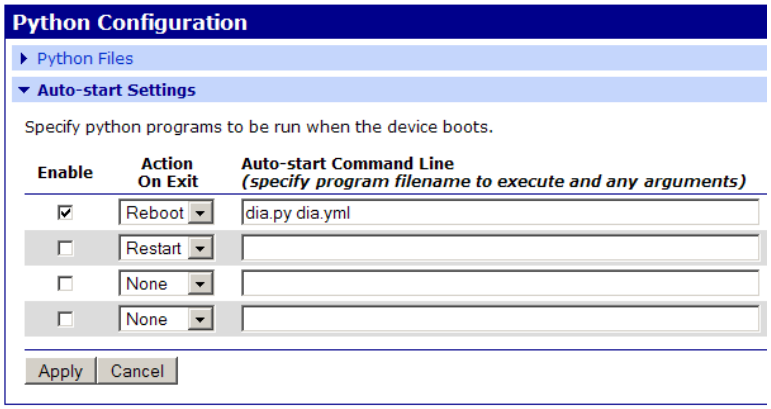
The files indicated here are required to set up and operate the M3 Sensors. They can be obtained from www.massa.com. The dia.yml file will need to be edited to configure the M3 Sensors to operate with your gateway. See Section 3.0 for more information.

The operating files require include:

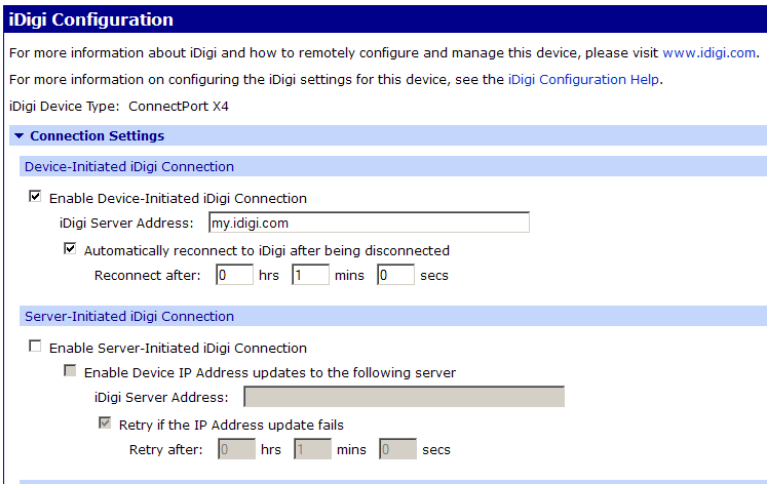
a) python.zip
b) zigbee.py
c) dia.py
d) dia.zip
e) dia.yml (This is used to control the operating mode when gateway boots. See section 3.1 for more information)



To have the gateway start executing the DIA (device integration application) when it powers up or reboots, check the *Enable box* in the Auto-start Settings UI. See section 3 for creating additional auto-start command lines as an option.



Here is the setting for sending the M3 automated message to the iDigi Cloud server. Contact Digi International for support.


iDigi Configuration
For more information about iDigi and how to remotely configure and manage this device, please visit www.idigi.com.
For more information on configuring the iDigi settings for this device, see the iDigi Configuration Help.
iDigi Device Type: ConnectPort X4
Connection Settings
Device-Initiated iDigi Connection
Enable Device-Initiated iDigi Connection
iDigi Server Address: my.idigi.com
Automatically reconnect to iDigi after being disconnected
Reconnect after: 0 hrs 1 mins 0 secs
Server-Initiated iDigi Connection
Enable Server-Initiated iDigi Connection
Enable Device IP Address updates to the following server
iDigi Server Address:
Retry if the IP Address update fails
Retry after: 0 hrs 1 mins 0 secs

3.0 Operating Modes for Gateway and Sensor

There are several operating modes for the gateway using the Massa python driver. The primary function is listening for automated messages from M3 Sensors and sending its data to a cloud or local server. Secondary functions include programming sensor registers, programming XBee registers, obtaining sensor diagnostic data, communicating with the M3 Software application, reprogramming sensor firmware, and reprogramming the XBee radio.

Sensors from the factory have their PAN IDs set to zero which allows them to associate to any gateway when permissible (gateway won't permit a sensor to attach if the gateway has reached its child table limit, for example). They are also programmed to stay awake when they find a gateway (indicated by Sensor LED blinking every 2 seconds). Note, if gateway is not powered or sensor is not able to attach to a gateway, then the sensor will search for gateway every 60 seconds and thus blink in this manner (when sensors that are not programmed for sleep).

3.1 Configuring Sensor Settings & Other Controls

The controls for the gateway operating mode are found in the dia.yml file which is processed by the Massa Python driver (m3_driver.py). This dia.yml can be set up to program sensor's registers, program Xbee's registers, obtain diagnostic information, program the sensor's application firmware, and program the XBee radio. After the dia.yml is processed when the gateway starts, it returns to a listening mode waiting for automated M3 messages from the sensors. The dia.yml also includes drivers to send this data to cloud or local servers (see Section 3.5). See the following pages for the definition of each setting. Shown below is an excerpt for the settings from a dia.yml[1] file (with default settings):

```
- name: M3-150-3b4e or M3-150-Tank1          # Sensor name by user[2]
  driver: devices.vendors.massa.m3_driver:M3Driver   # Massa M3 Sensor python driver
  settings:
  # Main dia related
      extended_address: '00:13:a2:00:40:48:3b:4e!'   # MAC Address of the M3 Sensor (must be lower case and followed by !)
      xbee_device_manager: 'xbee_device_manager'     # operating manager (required)
  # Sensor settings[3]
      sleep_interval_sec: 0                          # Deep Sleep Interval (seconds), limits: 0, 12 – 88474 seconds
      awake_time_sec: 31                             # Awake Time (seconds), limits: 12 – 600 seconds, 20 sec min for most appl.
      data_collection_interval_sec: 0                # Data Collection Interval (seconds), limits: 0, 10 – 16777215 seconds
      outgoing_message_mode: 0                       # Outgoing message mode, limits: 0-6
      auto_message_length: 8                         # Outgoing message length, limits: 1-8 (main sensor version 27 or greater)
      sensitivity: 'normal'                          # Sensor Sensitivity ('normal', 'high', 'very high', 'low', 'very low', 'custom[4]')
      user_comment: ' '                              # ASCII up to 32 characters (between quotes)
      comm_fail_boot: 0                              # Reboot sensor on failed Zigbee Ack / Unassoc. limits: 0, 4 – 254 awake periods
  # XBee settings[5]
      pan_id: 0x0000000000000000                     # PAN (Personal Area Network) ID limits: 0 - 0x000000007fffffff
      xbee_security: 'disable'                       # Security encryption enable (EE). To activate enter 'enable'
      xbee_security_key: 0x0000000000000000          # Security encryption key (KY), limits: 0 - 0xffffffffffffffff
      poll_rate: 0                                   # Poll rate register (PO), set to 0
      node_discovery: 1                              # Node discover register (NO), set to 1
      pan_conflict_threshold: 3                      # PAN conflict threshold register (CR), set to 3, limits 1 - 0x3f
      device_options: 0                              # Device options register (DO), limits 0 or 4, response 0=first, 4=best, ver 29A7 or greater
  # Routine control[6]
      clear_data_history: 'false'                    # Select 'true' to clear sensors historical data
      obtain_diagnostic_data: 'false'                # Obtains waveforms and other data upon gateway boot. To enable select 'true'
      program_m3_firmware_file: ' '                  # Update sensor firmware, file named here located in Python manage files.
      keep_xbee_awake: 'false'                       # 'true' to keep sensor awake 5 minutes, 'false' to disable this function
      clear_errors: 'false'                          # 'true' to clear sensors errors
      reboot_sensor: 'false'                         # 'true' to reboot sensor. Required after XBee firmware is updated
      log_status: 'false'                            # 'true' to output a 'status_log.txt' file that records autosend messages
      log_dia: 'false'                               # 'true' to output a 'dia_log.txt' file that reports sensor settings and driver version
```

Notes:

1. This example contains just part of the dia.yml file. See Appendix B for additional settings. The '#' character and all that follows on a line are comments and are not processed. Quote marks can be single or double as long as they match.
2. Suggested naming convention is to specify sensor model number followed by sensor MAC address and/or tank serial number.
3. Sensor registers listed in the dia.yml will cause writes to sensor registers only if they are different. Sensor registers are read when gateway boots (and after sensor wakes up). Registers left out of dia.yml will not be written.
4. Custom Sensitivity will require custom settings registers to be set, see Section 3.1.11 for these registers (not shown in this excerpt).
5. XBee registers listed in the dia.yml will write to Xbee registers. Registers left out of dia.yml will not be written.
6. dia controls listed in the dia.yml will execute specific functions when set to do so.

If a dia controls are left out or preceded with the '#' symbol, then the specific function will not be executed.

3.0 Operating Modes for Gateway and Sensor

3.1.1 Dia.yml Settings Description - *name*

The setting "name" in the dia.yml file is used to identify the sensor to the device server. A naming suggestion to best help identify a sensor in multiple tanks is to use the sensor model number followed by the sensor's MAC address or tank number or serial number as indicted. Limit: 32 characters, no quotes.

name: **M3-150-7A563B40** or **M3-150-7A563B40-Tank1** *(these are examples)*

3.1.2 Dia.yml Settings Description - *driver*

This is the python driver for the Massa M3 sensor "m3_driver" located in the dia.zip file.

driver: **devices.vendors.massa.m3_driver:M3Driver**

3.1.3 Dia.yml Settings Description – *Xbee extended address*

The setting "extended_address" in the dia.yml file identifies the sensor for the driver. Enter the entire MAC address as seen on the label on the sensor or as found in the discovery process. Use lower case for the alpha characters and end with an exclamation point and enclose with quotes. Add additional sensors by separating them by commas.

extended_address: **'00:13:a2:00:40:48:3b:4e!'**

3.1.4 Dia.yml Settings Description – *Xbee device manager*

The setting "xbee_device_manager" found in the dia.yml file is used to specify the Xbee manager. Quotes required.

xbee_device_manager: **'xbee_device_manager'**

3.1.5 Dia.yml Settings Description – *Sleep Interval*

The setting "sleep_interval_sec" in the dia.yml file controls the XBee radio sleep period. For Modes 1-3 the sensor will wake up radio to send status information and is independent of the rate to which data is acquired (see data_collection_interval_sec setting). For Modes 5 and 6, the sensor will obtain new range and status information, then wake up the radio and send it to the gateway. The recommended sleep interval settings are 1, 2, 4, 8, or 12 hours. See appendix for battery life estimates with different Sleep Interval settings. No radio communications can occur while sensor is sleeping (red LED off). Leaving this control line out of the dia.yml, blank value, or incorrectly typed setting value will not access this sensor setting. Limits: 0, 12 – 88474 seconds.

sleep_interval_sec: **0** (seconds)

NOTE: If this sleep setting is programmed to a non-zero value, program the outgoing_message_mode to non-zero as well.

3.1.6 Dia.yml Settings Description – *Awake Time*

The setting "awake_time_sec" in the dia.yml file controls the XBee radio ON time after the sleep timer has expired. Communications occurs during this period and will be indicated by the red LED blinking every 2 seconds. Recommended setting is 25 seconds. Leaving this control line out of the dia.yml, blank value, or incorrectly typed setting value will not access this sensor setting. Limits: 12 – 600 seconds.

awake_time_sec: **0** (seconds)

3.1.7 Dia.yml Settings Description – *Data Collection Interval*

The setting "data_collection_interval_sec" in the dia.yml file controls the rate at which an ultrasonic range measurement and other status information are collected. This setting is ignored when the Outgoing Message Mode register is set to 5 or 6 which uses the Sleep Interval to obtain range and status. For more information, see Section 3.1.8. Leaving this control line out of the dia.yml, blank value, or incorrectly typed setting value will not access this sensor setting. Limits: 0, 10 – 16777215 seconds.

data_collection_interval_sec: **0** (seconds)

3.0 Operating Modes for Gateway and Sensor

3.1.8 Dia.yml Settings Description – *Outgoing Status Message Mode*

The setting "outgoing_message_mode" in the dia.yml file sets the type of automated outgoing status message format when the sensor wakes up from sleep. See Appendix E for the format of the outgoing message and section 4.0 for an applications example. Leaving this control line out of the dia.yml, blank value, or incorrectly typed setting value will not access this sensor setting. Limits: 0 – 6 (Limits 0 – 4 for sensor main versions to 25)

outgoing_message_mode: **0**

outgoing_message_mode	mode name	Outgoing Status Message Mode Description
0	Manual	Sensor will only send messages upon request during the Awake period
1	Auto Send 1	Sensor will send a single Command 1 message containing up to 8 blocks of Event Records will wait for the entire programmed Awake Time for any normal protocol requests. There is no demand for any acknowledgement and no retry will be sent.
2	Auto Send 2	Sensor will send a single Command 1 message containing up to 8 blocks of recorded Event Records will wait for ½ the programmed Awake period then sensor will send a second copy. Awake period will continue in its entirety for any normal protocol requests. There is no demand for any acknowledgement and no retry will be sent.
3	Auto Send 3	Sensor will send a single Command 1 message containing up to 8 blocks of recorded Event Records will wait for ½ the programmed Awake period for a protocol acknowledgement (Command 200). If acknowledge message is not received within ½ of Awake period, then a second copy will be sent. Awake period will continue in its entirety for any normal protocol requests.
4	Auto Send 4	Sensor will send a single Acknowledge message indicating that it is awake. Coordinator now can send messages during the Awake period.
5	Auto Send 5	Sensor will first acquire range and other status information followed by sending a single Command 1 message containing up to 8 blocks of recorded Event Records. The Awake period will continue in its entirety for any normal protocol requests. There is no demand for any acknowledgement and no retry will be sent. The Data Collection Interval Register is disabled.
6	Auto Send 6	Sensor will first acquire range and other status information followed by sending a single Command 1 message containing up to 8 blocks of recorded Event Records. Sensor will wait for ½ the programmed Awake period then send a second copy. The Awake period will continue in its entirety for any normal protocol requests. There is no demand for any acknowledgement and no retry will be sent. The Data Collection Interval Register is disabled.

3.1.9 Dia.yml Settings Description – *# of Records in Outgoing Status Message*

The setting "auto_message_length" in the dia.yml file controls the number of Event Blocks (records) within the automated outgoing message when sensor wakes up from sleep. See Appendix E for the format of the outgoing message. Leaving this control line out of the dia.yml, blank value, or incorrectly typed setting value will not access this sensor setting. Limits: 1 – 8.

auto_message_length: **8**

3.1.10 Dia.yml Settings Description – *Ultrasonic Signal Sensitivity Setting*

The setting "sensitivity" in the dia.yml file controls the thresholds levels and timing for capturing the reflected signal. Most applications should have this set to 'Normal'. When considering settings other than Normal, set the dia.yml to obtain waveforms for a target placed at the furthest distance, like an empty tank. Review the waveform as seen in Appendix F to determine if this setting requires a setting other than Normal. Poor reflections should go in the direction of a Very High setting while very strong signals and applications with short sensing ranges can consider settings in the Very Low direction. Customized settings is an option on more difficult applications, consult Massa Products for support. Leaving this control line out of the dia.yml, blank value, or incorrectly typed setting value will not access this sensor setting. Select 'normal', 'high', 'very high', 'low', 'very low' or 'custom'. Enclose with quotes.

sensitivity: **'normal'**

3.0 Operating Modes for Gateway and Sensor

3.1.10 Dia.yml Settings Description – *Ultrasonic Signal Sensitivity Setting* (continued from previous page)

sensitivity	Description
Very Low	This setting should be considered if application has a short maximum sensing range.
Low	This setting should be considered if application has a short maximum sensing range.
Normal	This is the default setting to be used in most applications.
High	This setting should be considered if application requires sensing out to maximum specified range for sensor.
Very High	This setting should be considered if application requires sensing out to maximum specified range for sensor and/or liquid surface is know to foam.
Custom	Some applications may have nuisance reflections that may have any options of moving the nuisance or sensor mounting location (such as a ladder in a tank). Obtain diagnostic waveforms in an empty tank then consult Massa Products for support in setting up a custom setting for your application.

3.1.11 Dia.yml Settings Description – *Custom Ultrasonic Signal Sensitivity Settings*

The sensitivity settings listed here allow customization of the thresholds and timing for target detection on the ultrasonic signal for more difficult applications. These are active when the "sensitivity" setting is set to 'custom'. As described in section 3.1.10, most applications should use the sensitivity setting to 2 (Normal settings). These customized sensitivity values will be seen when you obtain the sensor's waveforms allowing them to be validated for your application. Leaving these control line out of the dia.yml, blank value, or incorrectly typed setting value will not access these sensor settings. Limits: see tables below.

```
sensitivity: 'Custom'                          # Sensor Sensitivity ('Normal', 'High', 'Very High', 'Low', 'Very Low', 'Custom')
custom_sensitivity_threshold1: 0               # Custom sensitivity threshold #1
custom_sensitivity_threshold2: 0               # Custom sensitivity threshold #2
custom_sensitivity_threshold3: 0               # Custom sensitivity threshold #3
custom_sensitivity_threshold4: 0               # Custom sensitivity threshold #4
custom_sensitivity_threshold2_msec: 0          # Custom sensitivity switch time #2
custom_sensitivity_threshold3_msec: 0          # Custom sensitivity switch time #3
custom_sensitivity_threshold4_msec: 0          # Custom sensitivity switch time #4
```

Available Threshold Voltages (for setting 'custom_sensitivity_thresholdx')

Index	Voltage	**Index**	Voltage	**Index**	Voltage	**Index**	Voltage
0	off	**6**	0.96	**12**	1.31	**18**	1.66
1	0.70	**7**	1.05	**13**	1.40	**19**	1.75
2	0.79	**8**	1.14	**14**	1.49	**20**	1.84
3	0.82	**9**	1.17	**15**	1.52	**21**	1.93
4	0.88	**10**	1.23	**16**	1.58	**22**	2.01
5	0.93	**11**	1.28	**17**	1.63		

Threshold Switch Time Limits (for setting 'custom_sensitivity_thresholdx_msec')

Sensor Model	Threshold Switch Time Limits (mS)
150 and 95	0.5465 – 24.8
50	1.57 – 109

3.0 Operating Modes for Gateway and Sensor

3.1.12 Dia.yml Settings Description – *User Comment*

The setting "user_comment" in the dia.yml file allows descriptive information to be programmed into the sensor. Up to 32 ASCII characters can be placed into the sensor. Leaving this control line out of the dia.yml, blank value, or characters other than ASCII will not access this sensor setting. Limits: up to 32 ASCII characters within quotes. A space or no space between the quote will program 32 spaces into these sensor comment registers. Enclose with quotes.

user_comment: ' '

3.1.13 Dia.yml Settings Description – *Zigbee Ack / Unassociated Failure to Boot Sensor*

The setting "comm_fail_boot" allows for fail safe operation by rebooting sensor in cases of consecutive failed Zigbee Acknowledge message or Unassociated state with gateway that occur on awake cycles. Suggested setting for a reboot is 24 hours and timing is based on the Sleep Setting (sleep_interval_sec). For example, if sleep setting is 4 hours, set this register to 6 for sensor to reboot on 6 consecutive awake cycles that cannot communicate with gateway (for 24 hours). Limits: 0, 4 – 254. "0" disables this feature. Invalid values (1, 2, 3 & 255) will be default to 0.

comm_fail_boot: **0**

3.1.14 Dia.yml Settings Description – *XBee PAN ID (ID)*

The setting "pan_id" in the dia.yml file programs the XBee's PAN ID register. Sensors are shipped from the factory with its PAN ID set to zero, which by definition, will commission to any gateway (any PAN ID). It is suggested that if more than one gateway and sensor system is used within RF range of each other, that each gateway and associated sensor(s) be programmed with their own unique non-zero PAN ID. Set this value to the same non-zero PAN ID as seen in your Gateway's XBee PAN ID. Leaving this control line out of the dia.yml, blank value, or incorrectly typed setting value will not access this radio setting. Limits: 0 - 0x000000007fffffff

pan_id: **0x0000000000000000**

NOTE: Once the sensor's PAN ID has been changed after rebooting the gateway in the "Configure Sensor Settings" mode, the gateway's PAN ID will need to be changed with this newly assigned PAN ID so that you can continue to communicate with the sensor. If the Deep Sleep register is zero, the sensor will default to a 1 minute sleep mode and may take up to 1 hour to rejoin the gateway. If the Deep Sleep register is non-zero and the Outgoing Message Mode is 1,2, 5 or 6, then sensor will rejoin upon the next wake up cycle. See Section 2.3 to change the gateway's PAN ID.

3.1.15 Dia.yml Settings Description – *XBee Encryption Enable (EE)*

The setting "xbee_security" in the dia.yml file is used to set the XBee's *Encryption Enable* (EE) register. To activate sensor Xbee security, enter 'enable' and enter a security value in the 'xbee_security_key' setting (see section 3.1.16). Leaving this control line out of the dia.yml, blank value, or incorrectly typed setting value will not access this radio setting. The gateway's XBee "Encryption Enable (EE)" register will need to be set to '1' as well. Select 'enable' or 'disable'. Enclose with quotes.

xbee_security: **'disable'**

3.1.16 Dia.yml Settings Description – *XBee Encryption Key (KY)*

The setting "xbee_security_key" in the dia.yml file is used to set the XBee's *Encryption Key* (KY) register. This key will be loaded into XBee if xbee_security: 'enable'. The gateway's XBee "Link encryption key (KY)" register will need to be set to the same value as you have set here. Leaving this control line out of the dia.yml, blank value, or invalid value will not access this radio setting. Limits: 16 byte hexadecimal string.

xbee_security_key: **0x0000000000000000**

3.0 Operating Modes for Gateway and Sensor

3.1.17 Dia.yml Settings Description – *XBee Poll Rate (PO)*

The setting "poll_rate" in the dia.yml file is used to set the XBee's *Poll Rate* (PO) register. Leaving this control line out of the dia.yml, blank value, or incorrectly typed setting value will not access this radio setting. Set this value to 0.

poll_rate: **0**

3.1.18 Dia.yml Settings Description – *XBee Node Discovery Register (NO)*

The setting "node_discovery" in the dia.yml file is used to set the XBee's *Node Discovery* (PO) register. Leaving this control line out of the dia.yml, blank value, or incorrectly typed setting value will not access this radio setting. Set this value to 1.

node_discovery: **1**

3.1.19 Dia.yml Settings Description – *XBee PAN Conflict Threshold Register (CR)*

The setting "pan_conflict_threshold" in the dia.yml file is used to set the XBee's *PAN Conflict Threshold* (CR) register. This setting is only available for XBee firmware releases 29A0 and greater. It is important to note that when upgrading XBee firmware from earlier versions than 29A0, that this register be programmed to the default value of 3. Leaving this control line out of the dia.yml, blank value, or incorrectly typed setting value will not access this radio setting. Limits: 1 - 0x03f. Set this value to 3 (default).

pan_conflict_threshold: **3**

3.1.20 Dia.yml Settings Description – *XBee Device Options Register (DO) (FW releases 29A7 or greater)*

The setting "device_options" in the dia.yml file is used to set the XBee's *Device Options* (DO) register. This setting is only available for XBee firmware releases 29A7 and greater. It is important to note that when upgrading XBee firmware versions prior to 29A7, that this device options register be updated. Setting to 0 will associate to gateway on first response. Setting it to 4 will associate to the gateway with strongest signal (when more than 1 gateway in range). Leaving this control line out of the dia.yml, blank value, or incorrectly typed setting value will not access this radio setting. Limits: 0 or 4. Set this value to 0 (default). After this setting has been updated, execute a sensor reboot. See *reboot_sensor* setting.

device_options: **0**

3.1.21 Dia.yml Settings Description – *Clear Data History Records*

The setting "clear_data_history" in the dia.yml file is used to clear all the sensors historical data records and will reset the Event Number to 1. Enter 'true' to enable this setting. Leaving this control line out of the dia.yml, blank value, or incorrectly typed setting value will not perform this function. Select 'true' or 'false'. Enclose with quotes.

clear_data_history: **'false'**

3.1.22 Dia.yml Settings Description – *Obtain Diagnostic Data (ultrasonic waveforms and historical records)*

The setting "obtain_diagnostic_data" in the dia.yml file is used to obtain diagnostic data which includes waveforms and other information that will be stored in gateway memory. Enter 'true' and boot gateway to acquire ultrasonic waveforms and other information which will be stored in the gateway. Leaving this control line out of the dia.yml, blank value, or incorrectly typed setting value will not perform this function. Select 'true' or 'false'. Enclose with quotes.

obtain_diagnostic_data: **'false'**

NOTE: If the sleep setting is not enabled, the gateway will restore the previously stored sleep setting after diagnostic data is obtained. If a gateway power cycle were to occur while obtaining diagnostic data, the sleep setting will be set to zero. It is recommended that if this setting is enabled, the 'sleep_interval_sec' also be enabled and value set to your application so it is properly restored.

3.0 Operating Modes for Gateway and Sensor

3.1.23 Dia.yml Settings Description – *Program Sensor Firmware*

The setting "program_m3_firmware_file" in the dia.yml file is used upload sensor firmware. The firmware will need to be loaded into the Python manage files memory prior to activation. Leaving this control line out of the dia.yml, blank value, or incorrectly typed filename or incorrect model number within filename will not perform this function. Values: firmware bin files, see www.massa.com. Enclose with quotes.

program_m3_firmware_file: **'M3-150V31-21.bin'** *(example)*

NOTE: If the sleep setting is not enabled, the gateway will restore the previously stored sleep setting after firmware programming. If a gateway power cycle were to occur during programming, the sleep setting will be set to zero. It is recommended that if this setting is enabled, the 'sleep_interval_sec' also be enabled and value set to your application so it is properly restored.

3.1.24 Dia.yml Settings Description – *Sensor XBee Keep Awake command*

The setting "keep_xbee_awake" in the dia.yml file will keep the sensor's XBee awake for 5 minutes so that XBee firmware can be uploaded. User must first go to Digi's "XBee Configuration - OTA Firmware Update Setup" UI and load the XBee firmware (ebl file). See section 3.2.6 for XBee firmware upload procedure. Leaving this control line out of the dia.yml, blank value, or incorrectly typed setting value will not perform this function. Select 'true' or 'false'. Enclose with quotes. NOTE: If this setting is enabled, then ALL OTHER settings within the yml are disabled for the sensor.

keep_xbee_awake: **'false'**

3.1.25 Dia.yml Settings Description – *Clear Sensor Errors*

The setting "clear_errors" in the dia.yml file will clear the sensor's error register. Select 'true' to clear all errors. Error codes are: bit0 = sensor configuration register replaced with default, bit1 = sensor detection fault, bit2 = temperature probe fault, bit3 = brown out fault, and bit4 = power supply low. Select 'true' or 'false'. Enclose with quotes.

clear_errors: **'false'**

3.1.26 Dia.yml Settings Description – *Reboot Sensor*

The setting "reboot_sensor" in the dia.yml file will restart sensor including resetting its XBee radio. This setting needs to be enabled Xbee firmware programming and should the only setting enabled. It is not necessary to reboot the sensor with any other sensor setting change. Select 'true' to enable this feature. Select 'true' or 'false'. Enclose with quotes.

reboot_sensor: **'false'**

3.1.27 Dia.yml Settings Description – *Status Output Logging of Sensor's Automated Messages*

The setting "log_status" in the dia.yml file will create a log file stored in the gateway's python files named "status_log.txt" that will record the sensor's automated status messages. Only the most recent record in the automated message will be recorded in this file. The file will append each new message. Use this feature for initial evaluation and disable it once the application has been validated since memory space is limited. Select 'true' to enable this feature. Enclose with quotes.

log_status: **'false'**

Sample output:

Sensor 00:13:a2:00:40:54:da:6a!: Dist=26.5" Temp=-3.6C Batt=4.8V UltrasonicStrength=75% RadioSignal=VeryStrong Event=377 2012-02-13 09:41:16

3.1.28 Dia.yml Settings Description – *Dia Log for Sensor Settings*

The setting "log_dia" in the dia.yml file will create a log file stored in the gateway's python files named "dia_log.txt" that will report the sensor's settings. Sleeping sensors will respond upon awake. This file will include the python driver version. Select 'true' to enable this feature. Enclose with quotes.

log_dia: **'false'**

NOTE: This feature will temporarily set sensor's sleep setting (sleep_interval_sec) to zero and then restore it to the value previously set in the sensor or if the sleep setting is enabled in the dia.yml, it will update to the new setting.

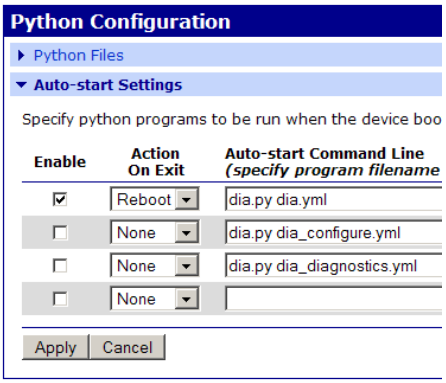
3.0 Operating Modes for Gateway and Sensor

3.2 Setting up gateway with multiple dia.yml

There are several ways to set up how the gateway will operate. One method is to use one dia.yml that is edited and reloaded for the different functions (configuring the sensor, obtaining diagnostics data - waveforms, firmware upload or a listening only mode for automated status messages). A second method is to write multiple yml files and give them their own unique name that is enabled in the "Auto-start Settings". Here it will be described the multiple dia.yml method versus editing the same dia.yml to perform different functions.

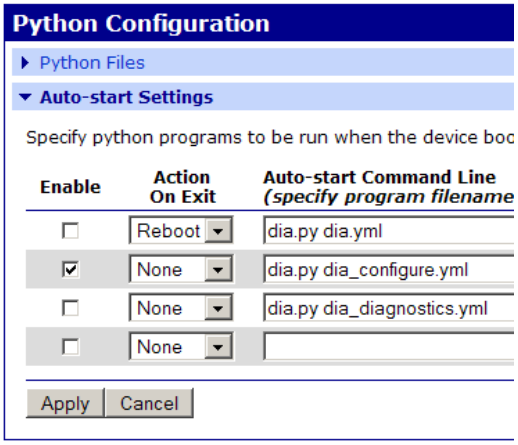
3.2.1 General Operation – No Action at Power Up (listen only)

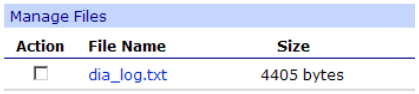
After the sensor has been configured for your application to automatically send its status message upon awake, you should set up a dia.yml so there are no settings that will be written to the sensor or XBee on subsequent power cycles or reboots. The dia.yml file can utilize the comment line indicator '#' for all controls. See example in Appendix B.



3.2.2 Configure Sensor settings with dia_configure.yml

To configure the sensors settings, edit the dia_configure.yml file for the desired sleep, awake, outgoing message mode, and other settings. Then load it into the python manage files. Check off "dia.py dia_configure.yml" in the Python "Auto-start Settings" UI. When gateway is booted, sensors settings will be updated when they wake up. For validating sensor settings, enable the setting log_dia: 'true'. The log file named "dia_log.txt" will be created in the python manage files. This file will contain sensor settings as seen in Section 3.1.26. Upon successfully changing the settings, revert back to listening only mode. See section 3.1 for configurable sensor settings. You may configure multiple sensors at a time with the yml file.





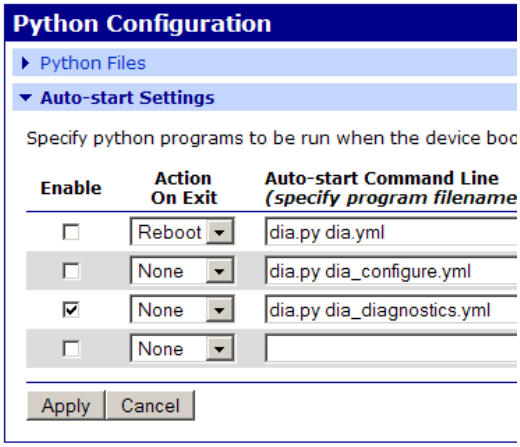
3.0 Operating Modes for Gateway and Sensor

3.2.3 Diagnostic Mode, Obtain Waveforms

To obtain diagnostic data that includes ultrasonic waveforms, sensor settings and history buffer up to the last 111 records, set up the command line in the dia.yml as:

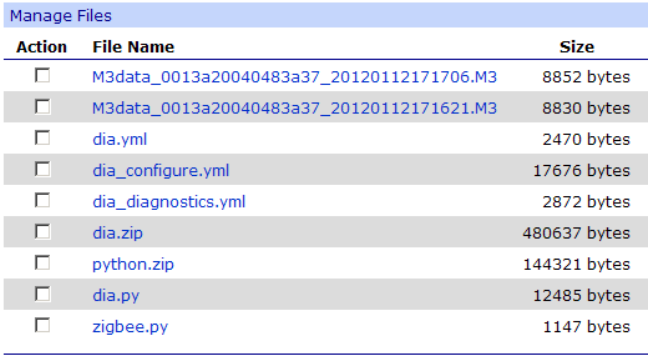
obtain_diagnostic_data: 'true'

This example has its own file named dia_diagnostics.yml and loaded into the Python Files. Enable the Auto-start Settings as shown and boot the gateway. When the sensor wakes up, it will be commanded to obtain waveforms. All waveforms collected will be placed in the *Python Configuration - Python Files - Manage Files* memory location. You may enable multiple sensors to obtain waveforms in the dia.yml.



NOTE: If the sleep setting is not enabled, the gateway will restore the previously stored sleep setting after diagnostic data is obtained. If a gateway power cycle were to occur while obtaining diagnostic data, the sleep setting will be set to zero. It is recommended that if this 'obtain_diagnostic_data' setting is enabled, the 'sleep_interval_sec' also be enabled and value set to your application so it is properly restored.

Shown here are 2 waveform files created from the M3/150 sensor (file extension ending in the .M3). As indicated in Appendix A, the M3/150 will obtain 2 waveforms while the M3/95 and M3/50 Sensors will obtain 4 waveforms. Download these files to your PC by clicking on the waveform file name which will open up a new window. Then save the waveform by selecting 'File' 'Save Page As' function. You may use the M3 Software to review the waveform (see Appendix F) by selecting 'File' 'Recall Sensor from File' function. The file name created by this diagnostic file contains the sensor's MAC address followed by the time stamp. Use this same name when saving the file.



After you have obtained the waveforms, you should disable obtain_diagnostic_data setting so subsequent gateway power cycles does not continue to request waveforms (see section 3.2.1). Delete these waveforms from the gateway after you have saved them due to gateway memory limitations.

3.0 Operating Modes for Gateway and Sensor

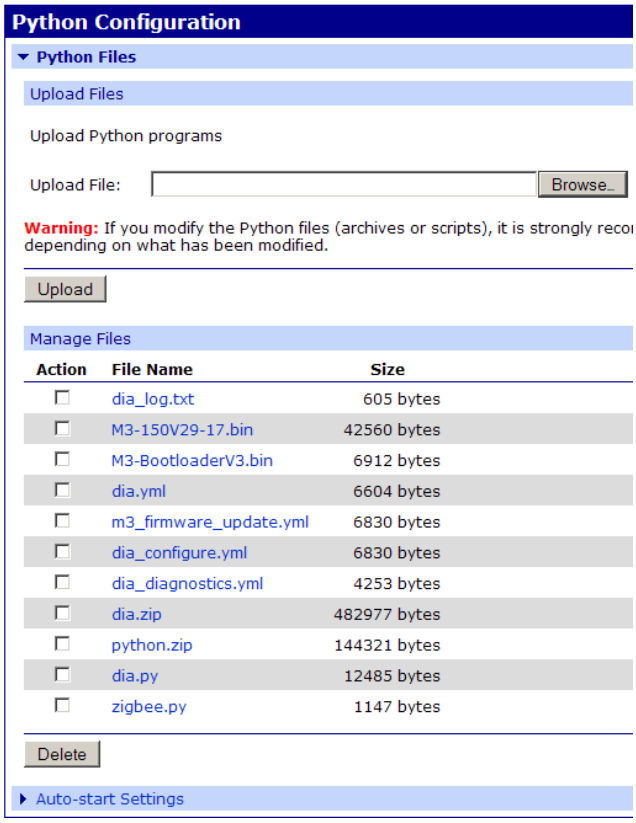
3.2.4 Sensor Application Firmware Update

First, obtain the sensor's application firmware from Massa Products web site and upload it as shown here stored in the *Python Configuration Manage Files* location. Firmware versions less than 29.x will also require loading M3-BootloaderV3.bin into gateway as seen here. Create a new yml file named *dia_m3_firmware_update.yml* and enable the program setting as shown here for model M3/150 sensor:

program_m3_firmware_file: 'M3-150V29-17.bin'

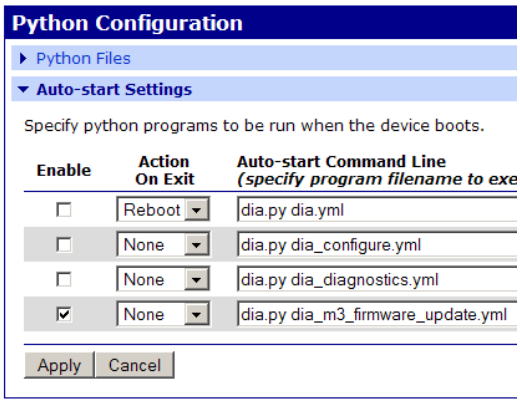
NOTE: <u>Enable only one sensor's program firmware setting at a time.</u>

NOTE: If the sleep setting is not enabled, the gateway will restore the previously stored sleep setting after firmware programming. If a gateway power cycle were to occur during programming, the sleep setting will be set to zero. It is recommended that if this setting is enabled, the 'sleep_interval_sec' also be enabled and value set to your application so it is properly restored.



In the Auto-start Settings menu, check the enable box for "dia.py dia_m3_firmware_update.yml" and reboot the gateway. When the sensor wakes up, it will begin uploading firmware. Upon completion, a dia log file named "dia_log.txt" will be created, or append to an existing one, and report if it has successfully uploaded the firmware.

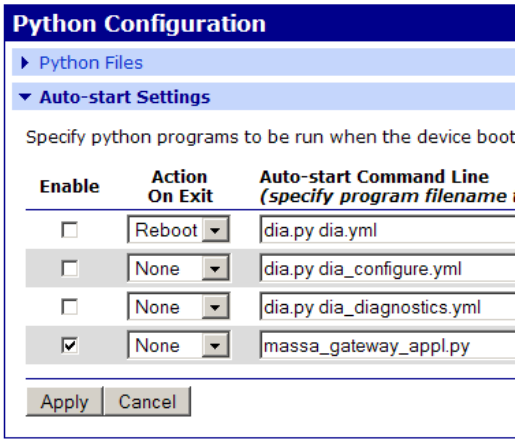
When successfully programmed, uncheck the Enable box for firmware uploading and return to the listening mode dia.yml (or another function) to prevent the sensor from having its firmware uploaded a second time on any subsequent gateway boots.



3.2.5 Enabling gateway serial port for use with M3 Software application (X4 Gateways only)

The M3 Software application is set up to communicate directly with an X4 Gateway's serial port. This is an alternate method of accessing sensors settings, utilizing the diagnostic tools, and update firmware locally. Load "massa_gateway_appl.py" then enable it in the Auto-start Settings UI. Reboot the gateway to enable the serial port.

Uncheck "Access the command line interface" in the Serial Services UI and set up for custom serial port setting: Baud Rate=9600, Data Bits=8, Parity=none, Stop Bits=1, and Flow Control=none. The M3 application software can be obtained at www.massa.com.



3.0 Operating Modes for Gateway and Sensor

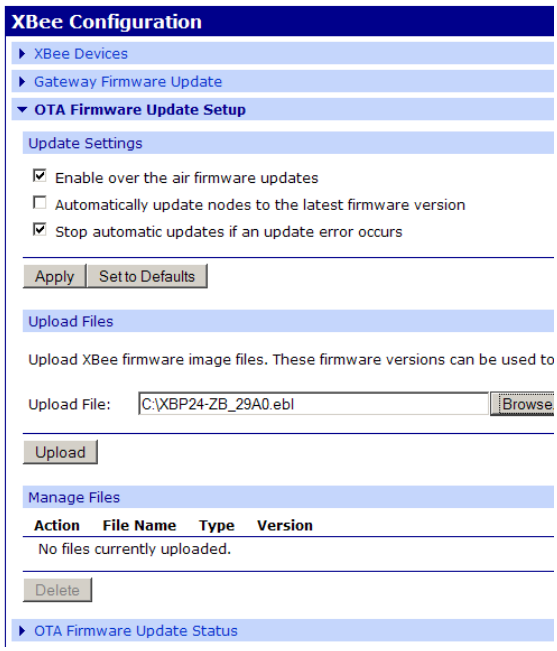
3.2.6 Sensor XBee Firmware Update

Follow the steps below to program sensors XBee firmware. You should obtain the firmware from either massa.com or from digi.com. Xbee firrmware has an "ebl" file extension. NOTE: Update only one Sensor XBee firmware at a time.

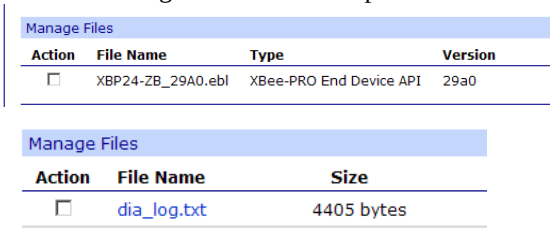
As of this writing, programming the XBee firmware requires several steps. It requires the sensor to keep its XBee ON for the entire programming process. This procedure is time sensitive because of the limitations of the radio which can only be instructed to stay awake for 5 minutes. The XBee programming takes about 1 minute.

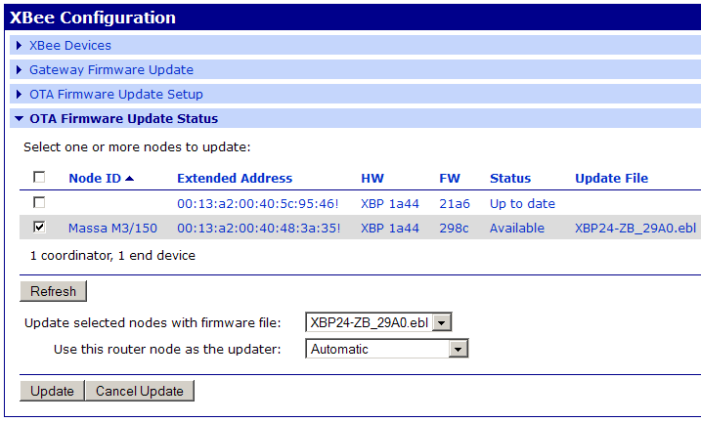
The gateway must be prepared to allow quick access to enable sensor XBee programming. Follow these steps to successfully program the sensors XBee. Enable setting *log_dia* and monitor dia_log.txt file for driver status for Step 6 below.

1) Verify that the sensor is awake so that it will be seen in the OTA Firmware Update Status UI. Boot gateway with dia_configure.yml setting *sleep_interval_sec: 0*. Also enable settings *poll_rate: 0* and *node_discovery: 1*.
2) In the *XBee Configuration – OTA Firmware Update Setup* UI, check the *Enable over the air firmware updates* and *Stop automatic updates if an update error occurs* then click Apply.
3) Browse for the latest XBee firmware (ver 29xx) and click Upload.
4) Go to the *OTA Firmware Update Status* UI and find sensor's XBee in the nodes list. If it does not appear on this list, then go to the *XBee Devices* UI and click on Discover Xbee Devices.
5) Boot gateway with dia_configure.yml setting *keep_xbee_awake: 'true'*. This will keep the sensor 100% awake for 5 minutes.
6) Monitor the dia_log.txt file in the python manges files UI and keep refreshing this log file until you get the "keep_xbee_awake message is acknowledged" message. When you receive this message with "True" indicated, proceed quickly to next step.
7) Immediately go to the *OTA Firmware Update Status* UI then check the sensor node, select node firmware ebl file, and click *Update*. Note Status will be indicated as *Updating*.
8) After 70 seconds, click on the *Refresh* button to get the indication that the XBee has updated the firmware with a *Complete* indication. This will validate that the XBee's firmware has been successfully updated.
9) IMPORTANT! After the sensor's XBee has been upgraded to ver 29A0 or greater, you must program a new XBee CR register (CR=3). Reboot gateway.
10) Lastly, reboot sensor by setting *reboot_sensor = 'true'*. Reboot gateway.
11) Disable *sensor_keep_awake, pan_conflict_threshold* and *reboot_sensor* settings after you verify sensor was rebooted.

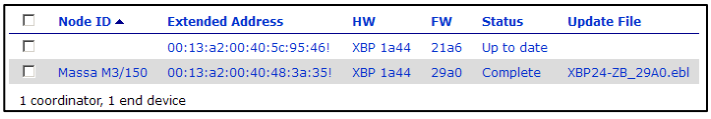


Manage Files after FW upload





XBee successfully programmed with FW level indicated



4.0 Application Setup Example

Application Example - Managing a tank farm

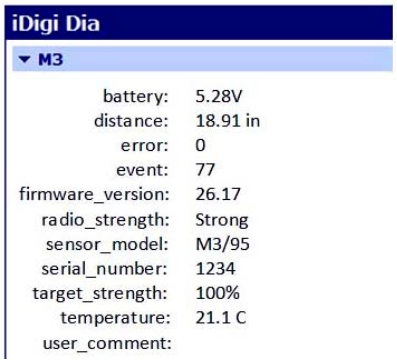
The M3 Sensors are supplied programmed to stay awake and will commission to any gateway (unless gateway security enabled or child table at max). This will be indicated by the sensor's red LED blinking every 2 seconds. Typical applications will have the sensors programmed for long sleep periods (1hr, 4hrs, 12hrs or more) to maintain battery life (see Appendix A). When programmed, sensors will awake and send status information to the gateway which in turn will send the information to a server (cloud or local). Setting up sensors and gateways to have their PAN ID's programmed to similar non-zero values will maintain that sensors stay associated with an assigned gateway and not join other gateways (when other gateways are in vicinity). Follow the steps below to properly set up your sensor and includes suggestions to obtain diagnostic data when evaluating a new tank for the very first time.

Procedure:

1) DO NOT POWER SENSORS AT THIS TIME.
2) Configure Digi Gateway as indicated by their instructions so you have an Ethernet connection. Use Digi's Discovery application to get to their Web UI.
3) If there are plans to ever have another gateway in the area, you should change the gateway's XBee Extended PAN identifier for the XBee network. See page 3.
4) Load all the python files from www.massa.com into your gateway.
5) Load a sample dia.yml (from www.massa.com) on your PC to edit the parameters as follows:
    a) Enter the MAC address of your sensor on the "extended_address" line to the same value as Step 3.
    b) Set other settings as seen in Appendix B
    c) If you have additional sensors, copy settings and edit MAC and NAME parameters. Also edit the PAN ID to the non-zero value the gateway was programmed to.
    d) Load the edited dia.yml back into the manage python files (see page 4)
    e) Power sensor now
    f) Re-boot gateway for pan id to be programmed into sensor. Sensors should stay commissioned going from PAN ID=0 to PANID=your new non-zero value.
    g) If evaluating a new tank, it is recommended to run diagnostics mode to obtain diagnostic data (ultrasonic waveforms) from the sensor. Data should be obtained with the tank empty, ½ full, and full. This data should be archived so that it can be used later to evaluate any problematic tanks. See section 3.2.3 for getting this data.
    h) To view waveforms, use the M3 Software Application, see Appendix F for a waveform description, but for support, contact Massa Products Corporation.
6) If the gateway was close by to the tank during setup and now you are ready to determine a permanent location of the gateway, you should program the sensor to send automated messages relatively fast (every few minutes). These messages will contain the RF signal strength. See Appendix C to observe the radio signal strength and where best to place the gateway as typically it will be more difficult to move the tank. Set up the dia.yml with the following settings:
    a) Obtain the dia.yml from gateway
    b) Set sample_rate_sec: 120
    c) Set sleep_rate_sec: 120
    d) Set awake_time_sec: 31
    e) Set outgoing_message_mode: 2
    f) Load the edited dia.yml back into the manage python files (see page 4)
    g) Re-boot gateway for settings to be programmed into sensor

4.0 Application Setup Example (continued from previous page)

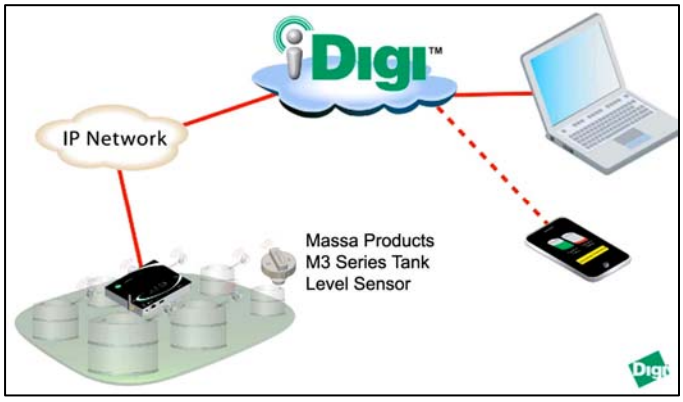
7) Go to the iDigi Dia web UI. This will be your gateway address followed by /idigi_dia. See figure to the right. Note: you must have the output drivers in the yml file as shown in Appendix B. An alternative way to obtain status messages is to enable the dia setting *log_status: 'true'* and view the status_log.txt file stored in the python files web UI. This setting stores the status information of the latest record of the automated message.



8) The RF signal strength report is named "radio_strength". If it is Moderate, Strong or Very Strong, then your gateway is mounted in a good location. If a report of "Weak" is reported, then review Appendix C for placement strategies. In some applications, you may have to move the sensor or place a wall router between sensor and gateway.

9) If your application requires security, you may program it as follows:
   a) Obtain the dia.yml from gateway
   b) Set xbee_security_enable: 'true'
   c) Set xbee_security_key: *your value*
   d) Repeat for all sensors in the dia.yml
   e) Boot gateway and wait for all sensors (depending on your current sleep setting) to be programmed with the security enable and keys.
   f) When all sensors have been programmed, update your gateway's XBee with Encryption enable (EE) = 1 and Link encryption key (KY) to save value you set the sensors. Click apply
   g) You will have to wait until sensor wakes up from its programmed sleep for it to re-associate with the gateway. If your sensor's sleep interval was programmed for 0, then you will have to wait up to one hour to re-associate. If sensor sleep setting was non-zero, then upon the next wake up cycle will it re-associate.

The gateway can be set up to route the sensor status information received to a server or cloud storage such as iDigi. Hosting sites can obtain and mange your data that could include sending you notification on tank level, alarms, and such to your PC or mobile device. For more information, contact Massa Products or Digi International.



APPENDIX A

Miscellaneous Sensor Information

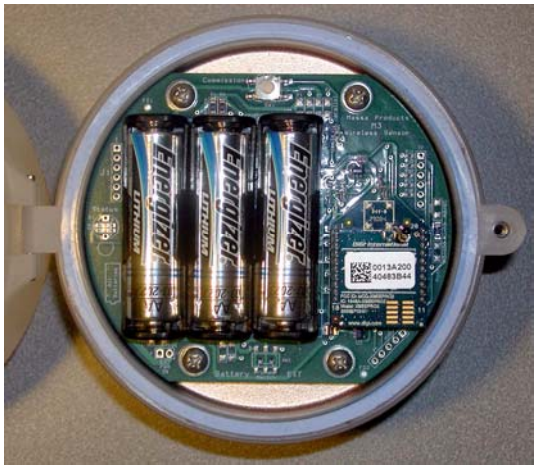
Battery Life and Suggested Replacement Voltage

Below are the battery life estimates using Energizer L91 batteries with the data acquisition rate programmed to the same time as the wake up time.

	Wake up and report status rate						
	24 hrs	12 hrs	8 hrs	6 hrs	4 hrs	1 hrs	1 min
Battery life at 80% battery efficiency	3.63 yrs	3.60 yrs	3.58 yrs	3.56 yrs	3.52 yrs	3.17 yrs	4.36 months

Status messages will include reporting the battery voltage. It would be a good idea to replace all three batteries when the sensor reports back with voltages below 3.5V. RF range becomes reduced with voltages below 3.2V.

To replace batteries, open cover using slotted screwdriver and remove existing batteries. Press commission button to discharge any residual voltage and then install batteries in polarity shown in battery holders, see photo to right below. Close and secure cover.



Diagnostic Mode – Sensor Waveforms

This table shows the waveform types that will be created and stored in the Manage Files memory location.

Waveform to save ➡	#1		#2		#3		#4	
Sensor Model	Transmit cycles	Gain	Transmit cycles	Gain	Transmit cycles	Gain	Transmit cycles	Gain
M3/150 and M3/150is	1	low	10	high	-	-	-	-
M3/95 and M3/95is	1	low	10	low	10	high	20	high
M3/50	4	low	1	low	10	low	10	high

APPENDIX A (continued from previous page)

Miscellaneous Sensor Information

LED Behavior

Following is the lookup table for the behavior of the LED:

LED Behavior	Description
OFF	No power, battery voltage too low, or sensor sleeping if powered
RED quick flash every 2 seconds	Sensor is joined with a ZigBee network, if stays in this mode, then Deep Sleep = 0.
RED quick flash every 2 seconds followed by an extended OFF period	Sensor is joined to a Zigbee network which then goes to sleep per programmed Awake setting and remains OFF for the programmed Deep Sleep setting.
RED 3 to 50 seconds steady ON	Un-associated state attempting to join ZigBee network.
RED quick flash once a minute	Sensor is not joined to a Zigbee network and the Deep Sleep Timer = 0. Once joined, sensor will quick flash every 2 seconds. May take up to 1 hour to join.
RED multiple blinking	Sensor was requested to read, write, obtain waveforms or other requests.
RED flash every ½ second for 30 seconds	Commission button pressed followed by other behaviors indicated above after 30 sec
GREEN 3 blinks followed by **RED**	Sensor was rebooted
RED 2 Hz flash, **GREEN** 1 Hz flash	Error: Application firmware not present

Rebooting Sensor Manually

To restart a sleeping sensor, remove battery and then press the commission button for several seconds. Reinstall battery.

Firmware Releases

As of this writing, gateway Xbee firmware is 21A0 and sensor is 29A0. See massa.com for the most up to date releases including sensor firmware.

Minimum Sensing Distance over Higher Temperatures

Temperature that exceed values indicated below will pose a slight variance in reporting a linear measurement when target approach the indicated minimum distances. The sensor's reported range will be the minimum specified distance when the target is at this distance.

Model M3/150

Temperature	Min Sensing Distance
< 25°C	4.25”
25°C to 40°C	4.5”
40°C to 50°C	4.75”
50°C to 60°C	5.0”
≥ +60C	5.25”

Model M3/95

Temperature	Min Sensing Distance
All	12”

APPENDIX A (continued from previous page)

Miscellaneous Sensor Information

Threshold Settings Defined in Sensor Registers (see section 3.1.10 and 3.1.11)

Model M3/150

Threshold Name	Reg 78	Reg 79	Reg 80	Reg 81	Reg 82-83	Reg 84-85	Reg 86-87
Very Hi	1	0	0	0	2400	3200	4000
Hi	4	3	2	1	2400	3200	4000
Normal	8	6	4	2	2400	3200	4000
Low	14	8	6	3	2400	3200	4000
Very Low	18	12	8	4	3200	4000	4800

Model M3/95

Threshold Name	Reg 78	Reg 79	Reg 80	Reg 81	Reg 82-83	Reg 84-85	Reg 86-87
Very Hi	1	0	0	0	8000	12000	16000
Hi	4	3	2	1	8000	12000	16000
Normal	8	5	2	1	8000	12000	16000
Low	14	6	4	2	8000	12000	16000
Very Low	18	12	7	3	8000	12000	16000

Model M3/50

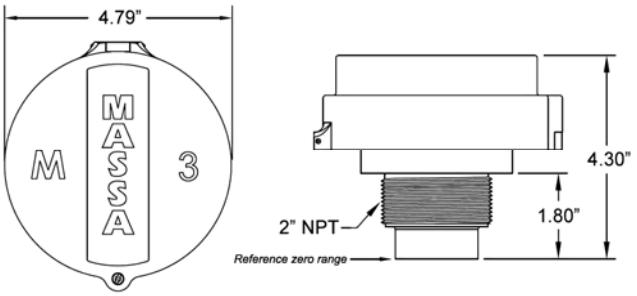
Threshold Name	Reg 78	Reg 79	Reg 80	Reg 81	Reg 82-83	Reg 84-85	Reg 86-87
Very Hi	10	7	4	1	8000	12000	16000
Hi	12	8	6	2	10000	14000	18000
Normal	14	9	6	2	12000	16000	20000
Low	19	13	7	3	12000	16000	20000
Very Low	22	17	11	6	16000	20000	24000

Sensor XBee radio settings (FW ver 29A0):

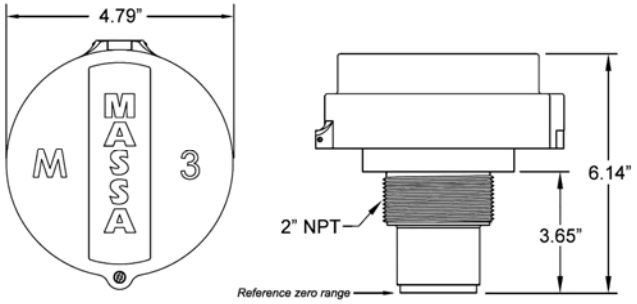
Scan Channels (SC) = 0x1ffe
Node Identifier (NI) = Massa M3/xxx (this is specific to sensor model type, 150, 95, 50, 150is, or 95is)
Device Type Identifier (DD) = 0x30201
Node Discovery Option (NO) = 1
Sleep Mode (SM) = 1 (Pin Hibernate)
Poll Rate (PO) = 0
PAN Conflict Threshold (CR) = 3
Device Options (DO) = 0 (future 29A7 or greater)
Associated LED Blink Time (LT) = 0x0A

Sensor dimensions:

Models M3/150 & M3/95



Model M3/50



APPENDIX B

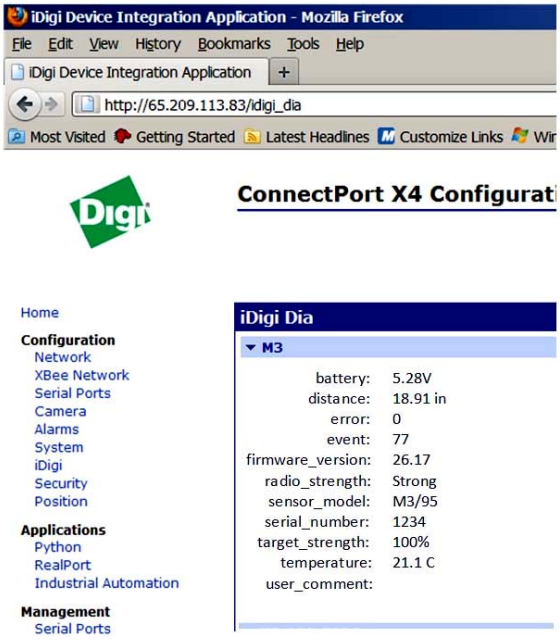
Format for dia.yml

The controls for the gateway operating mode are found in the dia.yml file which is processed by the Massa Python driver (m3_driver.py). See an example below. It is very important that the syntax is maintained with same spaces and no tabs as shown in this example. It is suggested that you download "Notepad ++" and set it up properly so that any tabs are converted to spaces. The "#" character and all that follows on a line are comments and are not processed. Quote marks can be single or double as long as they match. The driver below "loggers" is used for uploading data to the iDigi Cloud.

```
devices:
- driver: devices.xbee.xbee_device_manager.xbee_device_manager:XBeeDeviceManager
  name: xbee_device_manager
  settings:
    addr_dd_map: {}
    worker_threads: 1
##############################################################################################
# NOTE: If dia controls are left out or preceded with the '#' symbol (comment line), then the specific function
# will be ignored and not executed.  Enabling the xbee_program_awake setting will disable all other settings.
##############################################################################################
# Sensor #1 ##################################################################################
- name: M3-150-3b43
  driver: devices.vendors.massa.m3_driver:M3Driver
  settings:
# Main DIA related
    extended_address: '00:13:a2:00:40:48:3b:4e!'
    xbee_device_manager: 'xbee_device_manager'
# Sensor Settings
    # sleep_interval_sec: 0                          # units in seconds
    # awake_time_sec: 31                             # 20 sec min for most applications
    # data_collection_interval_sec: 0                # units in sec (inactive for outgoing msg modes 5 and 6)
    # outgoing_message_mode: 0                       # limit 0-6
    # auto_message_length: 8                         # limit 1-8
    # sensitivity: 'normal'                          # 'normal', 'high', 'very high', 'low', 'very low', 'custom'
    # user_comment: ''                               # limit 32 ASCII characters
    # comm_fail_boot:                                # Reboot sensor on failed Zigbee Ack / Unassoc. limits: 0, 4 – 254 awake periods, default = 0
# XBee settings
    # pan_id: 0x0000000000000000                     # limits 0x0000000000000000 to 0x000000007fffffff
    # xbee_security: 'disable'                       # XBee Encryption enable EE register 'enabled' or 'disabled'
    # xbee_security_key: 0x0000000000000000          # XBee (Link encryption key) KY register, 16 bytes
    # poll_rate: 0                                   # XBee PO register, PO=0 for XBee FW 29A0 or greater
    # node_discovery: 1                              # XBee NO register, 0 or 1, set this to 1
    # pan_conflict_threshold: 3                      # XBee CR register, limits 1-0x3f, set this to 3
    # device_options: 0                              # XBee DO register, limits 0 or 4. 0 for first find association, 4 for strongest signal association (XBee 29A7 or greater)
# Routine control
    # obtain_diagnostic_data: 'true'                 # 'true' to obtain waveforms or 'false' to disable. Recommended to enable sleep_interval_sec if this setting enabled.
    # clear_data_history: 'false'                    # select 'true' to clear sensor history
    # program_m3_firmware_file: 'M3-150V31-21.bin'   # see massa.com for most up to date sensor FW ver.  Recommended to enable sleep_interval_sec if this setting enabled.
    # keep_xbee_awake: 'false'                       # 'true' to keep sensor awake 5 minutes, 'false' to disable keep sensor awake function
    # clear_errors: 'false'                          # 'true' to clear sensor errors
    # reboot_sensor: 'false'                         # 'true' to reboot sensor.  Required after XBee firmware is updated
    # log_status: 'false'                            # 'true' to output a 'status_log.txt' file that records autosend messages
    # log_dia: 'false'                               # 'true' to output a 'dia_log.txt' file that reports sensor settings and driver version
##############################################################################################
# For additional sensors on the gateway, copy the sensor controls as seen in Sensor 1 above and edit the new control extended_address line with sensor's
# MAC address and assign a new name (-name: ).
loggers: []
presentations:
- driver: presentations.rci.rci_handler:RCIHandler
  name: rci
  settings: {}
- driver: presentations.idigi_db.idigi_db:iDigi_DB
  name: idigi_db
  settings:
    channels: []
    collection: ''
    compact_xml: false
    file_count: 20
    filename: channel_readings
    interval: 30
    sample_threshold: 20
    secure: true
- driver: presentations.console.console:Console
  name: console0
  settings: {baudrate: 115200, device: /com/0, port: 4146, type: tcp}
- driver: presentations.embedded_web.web:Web
  name: web0
  settings: {page: idigi_dia}
services: {}
```

APPENDIX C

Digi Gateway & Massa M3 Sensor Placement Strategies

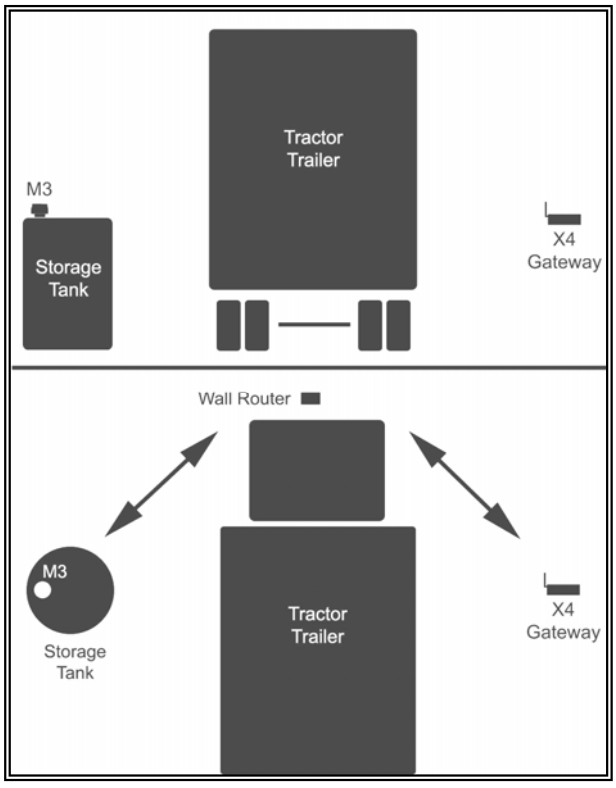


If you had been previously evaluating this kit in your lab environment, it is now time to move the gateway in a permanent location that will give the best possible wireless performance. Typically, the tank will be in a fixed location, so it will be necessary to place the gateway in a location where you will get the most reliable RF performance between sensor and gateway. This may even require the gateway to be placed outdoors in a weatherproof enclosure. To observe the RF signal strength, program the sensor to sleep every minute, awake for 30 seconds, and outgoing message to 1. Open the Web UI idigi_dia under your gateway's IP address as seen here. The RF signal strength is reported under "radio_strength:". Verify that both the sensor antenna and gateway antenna are vertical. *Moderate* reports are acceptable as long as the conditions don't change as you see in the following pages on placement strategies. *Weak* reports may require a change in gateway placement or the addition of wall routers. If there is a limitation on where the gateway can be placed, then a Digi Wall Router (repeater) should be used. Ultimately, it is best to keep hardware to a minimum (less wall router).

An alternative way to obtain status messages is to enable the dia setting *log_status: 'true'* and view the status_log.txt file stored in the python files web UI. This setting stores the status information of the latest record of the automated message.

NOTE: The gateway is rated for -30C to +70C and is not weatherproof. The sensor is watertight when securely tightened.

In this example, a tractor trailer truck is parked between a short storage tank and gateway. Radio signals may not be reliable.

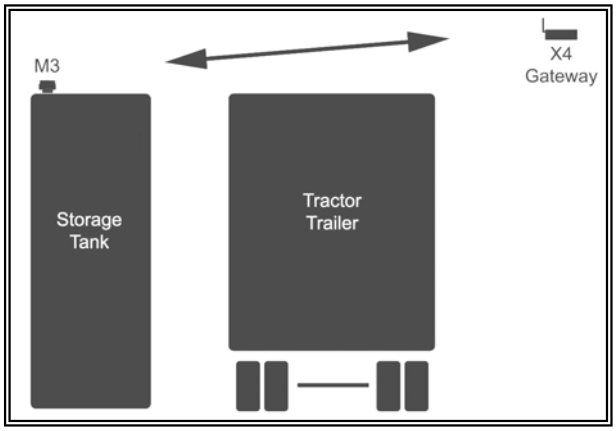


Placing a Digi Wall Router as shown here will allow another path around structures that could block radio signals.

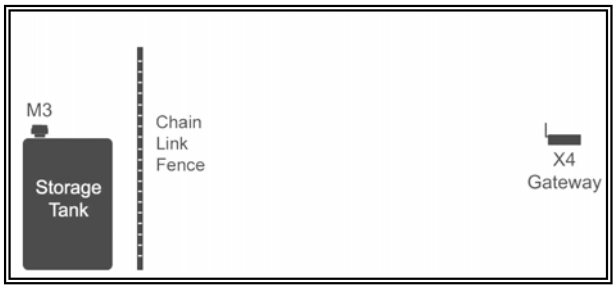
APPENDIX C (continued)

Digi Gateway & Massa M3 Sensor Placement Strategies

This example shows a much taller tank with a gateway placed high overhead. Here the tractor trailer does not impede the radio signals.



In this scenario, a chain link fence is between sensor and gateway. A metal fence will significantly reduce the radio range. It is better to place the tank with sensor on the other side of the fence or place a router in a location possibly over the fence.



This scenario will also reduce the effective radio range. Place wall routers between the 2 points or place the gateway closer to the storage tanks. The radio signals here can bounce away from the sensor if the roof is metal.

APPENDIX D

Fresnel zone and effects on it (see Digi web site for more info)

It makes sense that obstructions between a transmitter and receiver will reduce the communication range. In order to obtain the absolute maximum communication range possible, a radio modem system must be installed such that true RF Line-of-Sight (LOS) conditions exist between the transmitting and receiving antennas.

RF LOS is different that visual LOS. Visual LOS is present when one can stand next one antenna and use binoculars to view the other antenna. RF LOS requires not only a visual sight line between the antennas but it also requires that a football shaped area between the two antennas be free of obstructions.

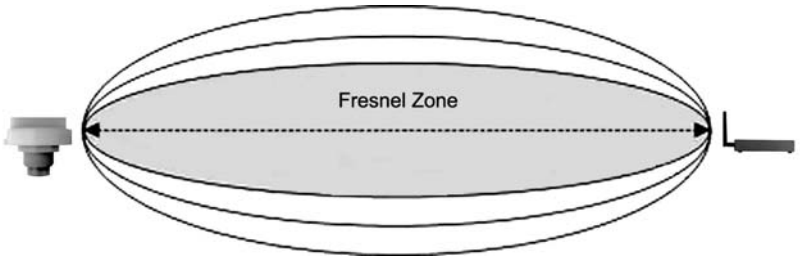
This football shaped area is called the Fresnel Zone (pronounced fernel zone). The Fresnel zone is an area that is larger in diameter at the center and smaller in diameter at either end. Also, the greater the distance between the antennas, the larger the diameter of the Fresnel zone in the center.

Any obstructions that enter into the Fresnel zone will reduce the communication range; including buildings, vegetation, the ground, etc. As the antennas get further apart and the diameter of the Fresnel zone increases, the ground can begin to obstruct the Fresnel zone. In order to keep the entire Fresnel zone free of obstructions it is necessary to raise the antennas. To keep the Fresnel zone off the ground the heights of the antennas added together must total more than the diameter of the Fresnel zone at the specific distance.

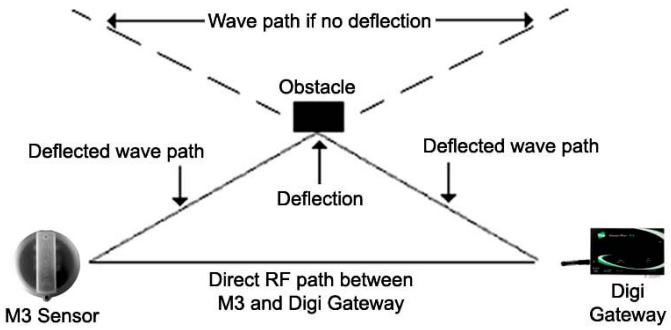
The diameter of the Fresnel zone is a function of the frequency and the distance between the antennas. For reference here is a table to use as a guideline:

	2.4 GHz	
Distance between antennas	**Fresnel zone diameter**	**Freespace loss (dB)**
1000 ft (300 m)	11 ft (3.4 m)	90
1 Mile (1.6 km)	21 ft (6.4 m)	104

Side view of Fresnel Zone



Since the RF signal is omni-directional, it can reflect signals from objects back to the sensor and gateway. This overhead view shows the RF signal deflecting off of an object that may reduce the signal of the direct RF path if out of phase. This example could be the source of problems of obtaining good signal strength between the 2 devices and is why it is important to perform site testing where the gateway and sensor it to be located. Most cased the reflected signal would be low in sensitivity and may not be an issue.



APPENDIX E

Outgoing Auto-Message Message Format

This is the automated outgoing message format when the sensor wakes up from Deep Sleep. The auto_message_length setting will be the same number indicated in the "EventsToRetrieve" byte. This setting allows you to reduce the size of this message by reducing the amount of records for such reasons as limiting the size for cell plan costs.

<Destination ID><Sender ID><Message Length><Command><Addr Ptr><EventsToRetrieve><Event Record (1)><Event Record (n)><Checksum>

Host ID	Sensor ID	total # of bytes	1	Record Pointer	# of event blocks	oldest record	newest record	checksum
default=251	default=1	of this message		1 to 111	1 to 8	(8 bytes)	(8 bytes)	

<u>Event Record (n)</u> block defined (8 bytes):

<EventLSB><EventMSB><Status1><Status2><RangeLSB><RangeMSB><TemperatureReading><BatteryReading>

<EventLSB><EventMSB>: 2-byte Event Counter will automatically increment for each new data acquisition

<Status1>: This status byte is parsed as;

<bit7> Sensor Error; 0=OK, 1=Fault (read Configuration Register 65)
<bit6> <bit5> undefined, factory only
<bit4> Ultrasonic Gain Control (short); 0=***Low***, 1=***Hi***
<bit3><bit2> Radio Signal Strength (of Last Reception); 00=***weak***, 01=***moderate***, 10=***strong***, 11=***very strong***
<bit1><bit0> Ultrasonic Signal Target Strength; 00=***<25%***, 01=***50%***, 10=***75%***, 11=***100%***

<Status2>: This status byte is parsed as;

<bit7><bit6><bit5> Ultrasonic Sensitivity Settings; 000=***Very Low***, 001=***Low***, 010=***Normal(default)***, 011=***Normal-Hi***, 100=***Hi***, 101=***Very Hi***, 110=***Custom***
<bit4><bit3> Ultrasonic Gain Control (long); 0=***Low***, 1=***Hi***, 2=***TimeVarying***
<bit2> Temperature Probe; 0=***Internal (default)***, 1=***User Programmed value)***
<bit1> Min Distance Processing; 0=***Disabled***, 1=***Enabled (default)***
<bit0> Range Bytes Resolution; 0= ÷128 (models M3/150 & M3/95), 1= ÷64 (model M3/50)

<RangeLSB><RangeMSB>: Ultrasonic Range (in inches). Resolution for model M3/150 and M3/95: RangeMSB:RangeLSB ÷ 128. Model M3/50: RangeMSB:RangeLSB ÷ 64. **Status2** <bit 0> indicates resolution. No target detected (no echo) is indicated with RangeMSB:RangeLSB=0. RangeMSB = 255 indicates range not acquired (not available in data history buffer).

Temperature (ºC) = 0.587085 * <TemperatureReading> - 50

Battery Voltage (volts) = (**<BatteryReading>** - 14) / 40

**** Reminder:** Resolution of the RangeLSB and RangeMSB is defined by Status2 bit 0.*

APPENDIX F

M3 Application Software – Understanding an Ultrasonic Waveform

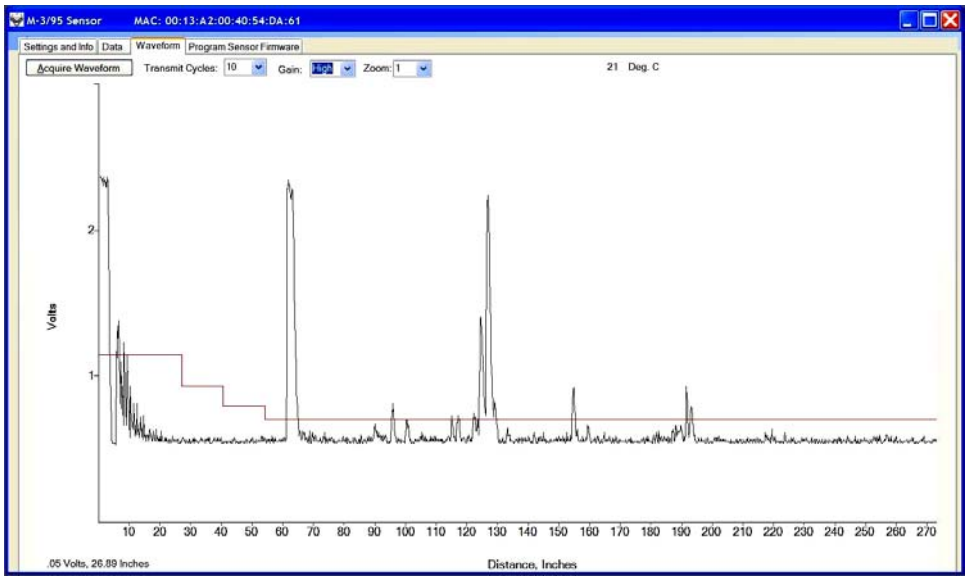
This very important tool will allow diagnosing more difficult applications and also allows recording the tanks ultrasonic profile for archival purposes.

When the dia.yml configures obtain_diagnostic_data_enable: 'false', waveforms and other data are stored in the gateway. Save the file from the gateway to your PC and use the M3 Application software (see www.massa.com) to recall each waveform. See plot below of a typical waveform.

A waveform can be broken down in 3 sections. The 1st pulse on the left in the waveform is from the sensor ping (or ultrasonic transmit) that emits the sound wave. It is then followed by a reflection (next pulse to the right of the ping pulse) off the surface and finally followed by other multiple reflections that may or may not be present. The sensor is designed to ignore the ping pulse and capture the 1st reflected signal over threshold to report the range to the surface (the red line in the waveform represents the signal capture threshold level). Other multiple reflections are ignored. Ideally the signal between the transmit pulse at the beginning of the waveform and the 1st reflected signal of the actual surface level needs to be clear of unwanted reflections. Unwanted reflections could be caused by items such as ladders in a tank, the sensor being mounted right next to the edge of the tank or the sensor not being level and pointing towards the edge of the tank or other items. In more difficult applications, the Sensor Sensitivity adjustments could provide a solution, but consult with Massa Products for support (record and provide all waveforms as indicated in Table 1 with an empty tank). As you can see in the waveform below, the 1st reflection is at 60" and it is clear from the ping pulse to this reflection (good!). An example of a problematic situation is for a pulse being present at 40" which could represent an item such as pipe in the direct path. Sound will travel around objects such as pipes for the bigger reflection such as the liquid surface.

If it is observed that a sensor model does not have a reliable reflection (1.25V peak at least) in an empty tank (or lowest process level) while operating with the highest power mode, then it may be necessary to select a lower frequency model sensor. Consult with Massa Products for more information.

NOTE: Model M3/150 has 2 waveforms as seen in Appendix A, all other models have 4 waveforms.



NOTE: The iDigi Device Integration Application (Dia) software provides a rich python library for addressing remote data acquisition, control, and presentation of device data on ConnectPort X gateways. Dia collects data from any device which can communicate with a Digi gateway, over any gateway interface and re-presents this data to upstream applications in a fully customizable approach. Digi recommends that you start your device networking application with Dia.

APPENDIX G

Checklist for Reliable Sensor and Gateway Operation

☐ Sensor XBee firmware version 29A0, PO=0, CR=3. Scan channel (SC) set to 0x1ffe.

a) PAN ID = 0 for fixed sensor applications that don't ever expect another gateway in the vicinity or for roving sensors that need to associate with any gateway.

b) PAN ID = non-zero for fixed sensor applications that do expect, or have another gateway in the vicinity, in which the sensor is associated is to a specific gateway.

☐ Gateway XBee firmware version 21A0, poll rate setting dependant on application

a) Fixed sensor placement such as tank farm – SN=2880, SP=1000 (24 hours). If sensor set up to sleep 12 hours or more, set for at least 48 hour period or SN=5760, SP=1000.

b) Roving sensors that don't need to stay associated to gateway – SN=3, SP=1000 (30 seconds)

c) Scan channel (SC) set to 0x1ffe

d) PAN ID = 0 will randomly assign an operating PAN ID. It should not be used if more than one gateway is in the vicinity in fixed sensor applications.

e) PAN ID = non-zero for fixed sensor applications that do expect, or have another gateway in the vicinity, in which the sensor is associated is to a specific gateway.

☐ Gateway firmware version 2.14.1.6 (Version 82001536_K 10/18/2011) or greater.

☐ Sensor settings:

a) Sleep setting to non-zero value which will output automated messages upon awake

b) Awake Timer is recommended for 31 seconds initially for setup and evaluating period. Minimum 20 seconds once application has been validated if desired.

c) Data Collection Interval set to your application. This is disabled for Outgoing Message Modes 5 & 6

d) Outgoing Message Mode – suggested 2 or 6 so that 2 messages will be sent upon awake in case of situations that one of the messages doesn't arrive due to a RF collision or other problem. Use Mode 5 or 6 to synchronize the sensing rate with sending the message.

e) Auto message length set your application.

f) Ultrasonic sensitivity set to normal. Consult factory for other settings in more difficult sensing applications.

☐ Massa python driver version 1.0 or greater. For version number, see log file dia_log.txt when dia executed.

☐ Application program "Notepad ++" is a very helpful in making sure your dia.yml has the proper spaces and tabs set to identify the controls to the dia. Make sure preferences are enabled for Python. Check and obtain from the web.

☐ dia.zip tested using DIA version 1.4.14

APPENDIX H

Troubleshooting Guide

Issue	Condition	Solution or work around
Missing data, sensor lost association.	- Gateway power cycle could lose association. - Gateway Xbee version less than version 21A0. - Poor RF link - Xbee firmware was updated	- Upgrade sensor firmware to at least version 31 which will detect unassociated state and attempt to rejoin gateway. - Upgrade gateway Xbee firmware to version 21A0. Sensor Xbee 29A0. - Consider the use of a wall router. Otherwise, relocate gateway. - Sensor antenna not upright - Poor sensor/gateway placement - low battery voltage
Sensors do not rejoin gateway upon awake when it was previously joined	Gateway power cycle and Xbee "Number of remaining children" register full (NC=0)	Upgrade Massa Python driver 0.53 or greater which will perform a Xbee network reset upon gateway boot when NC=10. Sensor firmware must be ver 31.x or greater which will have the sensor immediate rejoin the gateway upon its next awake cycle.
Cannot upload sensor firmware	Gateway XBee version 21A7 and encryption is enabled	Disable encryption (EE=0) in both sensor and gateway's Xbee. If encryption required (EE=1), then upload gateway Xbee version 21A0.
Sensor appears joined with RED LED flashing every 2 seconds (when awake) but not seen on XBee network). Case #1	Sensor associated with another gateway when several gateways operating in same vicinity (with sensor PAN ID=0)	Use non-zero PAN ID's for gateway and sensor.
Sensor appears joined with RED LED flashing every 2 seconds (when awake) but not seen on XBee network). Case #2	Gateway encryption enabled with sensor encryption disabled.	Sensor will be allowed to associate with the gateway but will not be seen or enabled on the XBee network.
Driver not outputting data or changing sensor settings	Using dia.yml controls	Check syntax and spacing for sensor settings in the dia.yml. Also check MAC address is correctly entered.
Sensor does not join gateway	RED LED is on solid and does not blink.	Gateway out of range, poor RF signal strength, Gateway XBee child table full (NC=0), gateway/sensor PAN ID don't match when non-zero for sensor, gateway/sensor's encryption key don't match, gateway SC register 0x3fff (set to 0x1ffe).
Sensor has to rejoin gateway every awake cycle	RED LED is on solid then blinks upon awake. This indicates gateway has dropped sensor from its XBee network.	Set gateway's SN/SP registers timing at least 3 times longer than the sensor's sleep register. Some applications will require gateway to intentionally drop sensors, so this will be normal LED operation as indicated in the "Condition" column.
Sensor blinking red/green.	No application firmware	Load gateway with M3 firmware (sensor model specific) and enable sensor firmware upload setting in dia.yml
M3/50 Sensor range reporting ½ the distance	Outgoing message range that is reported to server or cloud	Reminder that Status2 bit 0 indicated range resolution. See Appendix E.

VYDAS Vydas International Marketing
Swan House PBC Lynchborough Rd
Passfield Hampshire GU30 7SB UK
Tel: 44(0)1428 751822 Fax: 44(0)1428 751833
Email: info@vydas.co.uk Web: www.vydas.co.uk